

 **ONE IDENTITY™**

Syslog-ng 4 – Where log management is heading

Peter Czanik

Overview

- What is syslog-ng / logging
- Syslog-ng 3 retrospective
- What is new in syslog-ng 4
- Upgrading from version 3 to 4

What is syslog-ng?

- Logging: recording events, for example:

```
Jan 14 11:38:48 linux-0jbu sshd[7716]: Accepted  
publickey for root from 127.0.0.1 port 48806 ssh2
```

- syslog-ng: an enhanced logging daemon with a focus on portability and high-performance central log collection. Originally developed in C.

Freeform log messages

- Most log messages are: **date + hostname + text**

```
Mar 11 13:37:56 linux-6965 sshd[4547]: Accepted  
keyboard-interactive/pam for root from 127.0.0.1 port  
46048 ssh2
```

- Text = English sentence with some variable parts
- Easy to read by a human
- Difficult to create alerts or reports

Solution: structured logging

- Events represented as name-value pairs. For example, an ssh login:
app=sshd user=root source_ip=192.168.123.45
- syslog-ng: name-value pairs inside
 - Date, facility, priority, program name, pid, etc.
- Parsers in syslog-ng can turn unstructured and some structured data (CSV, JSON) into name-value pairs

Syslog-ng 3 retro

- Many features taken now for granted arrived in version 3:
- Disk-buffer
- Most of message parsing
- Python support
- HTTP destination
- Kubernetes support

Disk-buffer

- Temporary storage while a destination unavailable
- Normal vs. reliable:
 - Normal is reliable 😊
 - Reliable is slow and resource hungry

Message parsing

- Creates name-value pairs from interesting message parts
- PatternDB, CSV, JSON, XML, key=value, etc.
- Combined parsers in SCL: sudo, Apache, Cisco, etc.
- Application Adapters: automatic message parsing on the system() source
- Name-value pairs help in filtering, storage, and later in reporting, alerting, dashboards

Python support

- Originally for destinations, but now available everywhere
- Not just for PoC, Python is production ready
- Easy way to extend syslog-ng: no development tools or compiling software is necessary
- Easier to distributed your own extensions

HTTP destination

- `http()` is DIY
- In SCL `http()` is wrapped in many drivers:
 - Elasticsearch: `elasticsearch-http()`
 - Logging as a Service: Loggly, SumoLogic, etc.
 - Alerting to chat: Slack, Telegram, etc.
- Submit your integrations to be included in SCL

Initial Kubernetes support

- Can find and follow Kubernetes log files on the filesystem

Syslog-ng 4 is now available

- Version 4 of syslog-ng now available
- Fully backwards compatible in version 3 mode
- Runtime type information of name-value pairs in version 4 mode
 - From JSON and PatternDB parsers, rewrite rules
 - Can be used for comparisons, type aware output
- Improved list (array) handling
- sudo 1.9.4 JSON logs help to understand

What changed in type support?

- Syslog-ng still stores data as text, but version 4 now stores JSON/PatternDB parsed type information along with name-value pairs
- You can also set type manually
- You can use those for numerical comparisons (1.0 is no anymore equal to 1.1)
- Some destinations and template functions (JSON) can emit logs with proper type information

Syslog-ng configuration for JSON sudo logs

```
@version:3.37
@include "scl.conf"
source s_sys { system(); internal(); };
destination d_mesg { file("/var/log/messages"); };
log { source(s_sys); destination(d_mesg); };
filter f_sudo {program(sudo)};
destination d_test {
    file("/var/log/sudo.json"
    template("${format-json --scope nv_pairs --scope dot_nv_pairs --rekey .* --shift 1 --exclude *journal* --exclude MESSAGE --scope rfc5424}\n\n"));
};
log {
    source(s_sys);
    filter(f_sudo);
    destination(d_test);
};
```

Unparsed JSON sudo log

- Mar 17 15:24:07 localhost sudo[35095]:
@cee:{"sudo":{"accept":{"uuid":"9045212067-d7cb-4a8f-ab40-32dc5b7c6f","server_time":{"seconds":1679063047,"nanoseconds":415866617,"iso8601":"20230317142407Z","localtime":"Mar 17 15:24:07"},"submit_time":{"seconds":1679063047,"nanoseconds":399116920,"iso8601":"20230317142407Z","localtime":"Mar 17 15:24:07"},"submituser":"czanik","command":"/usr/bin/ls","runuser":"root","runcwd":"/home/czanik","ttyname":"/dev/pts/0","submithost":"localhost.localdomain","submitcwd":"/home/czanik","runuid":0,"columns":118,"lines":60,"runargv":["ls","/root/"],"runenv":["LANG=en_US.UTF-8","HOSTNAME=localhost.localdomain","TERM=xterm-256color","PATH=/home/czanik/.local/bin:/home/czanik/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin","MAIL=/var/mail/root","LOGNAME=root","USER=root","HOME=/root","SHELL=/bin/bash","SUDO_COMMAND=/usr/bin/ls","SUDO_USER=czanik","SUDO_UID=1000","SUDO_GID=1000"]}}}}

Log parsed and recreated by syslog-ng 3.X

```
{"cee":{"sudo":{"accept":{"uuid":"9045212067-d7cb-4a8f-ab40-32dc5b7c6f","ttyname":"/dev/pts/0","submituser":"czanik","submithost":"localhost.localdomain","submitcwd":"/home/czanik","submit_time":{"seconds":"1679063047","nanoseconds":"399116920","localtime":"Mar 17 15:24:07","iso8601":"20230317142407Z"},"server_time":{"seconds":"1679063047","nanoseconds":"415866617","localtime":"Mar 17 15:24:07","iso8601":"20230317142407Z"},"runuser":"root","runuid":"0","runev":"LANG=en_US.UTF-8,HOSTNAME=localhost.localdomain,TERM=xterm-256color,PATH=/home/czanik/.local/bin:/home/czanik/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin,MAIL=/var/mail/root,LOGNAME=root,USER=root,HOME=/root,SHELL=/bin/bash,\"SUDO_COMMAND=/usr/bin/ls /root/\",\"SUDO_USER=czanik,SUDO_UID=1000,SUDO_GID=1000\",\"run_cwd\":\"/home/czanik\",\"runargv\":\"ls,/root/\" \"lines\":\"60\" \"command\":\"/usr/bin/ls\" \"columns\":\"118\"}}}, \"app\":{\"name\":\"cee\"} \"SOURCE\":\"s_sys\" \"PROGRAM\":\"sudo\" \"PRIORITY\":\"notice\" \"PID\":\"35095\" \"HOST_FROM\":\"localhost\" \"HOST\":\"localhost\" \"FACILITY\":\"authpriv\" \"DATE\":\"Mar 17 15:24:07\"}}
```

Log parsed and recreated by syslog-ng 3.X

```
"runuser": "root",  
"runuid": "0",  
"runenv": "LANG=en_US.UTF-  
8,HOSTNAME=localhost.localdomain,TERM=xterm-  
256color,PATH=/home/czanik/.local/bin:/home/czanik/bin:/usr/local/bin:  
/usr/bin:/usr/local/sbin:/usr/sbin,MAIL=/var/mail/root,LOGNAME=root,U  
SER=root,HOME=/root,SHELL=/bin/bash,\"SUDO_COMMAND=/usr/bin/l  
s /root/\",SUDO_USER=czanik,SUDO_UID=1000,SUDO_GID=1000",  
"runcwd": "/home/czanik",  
"runargv": "ls,/root/",  
"lines": "60",  
"command": "/usr/bin/ls",  
"columns": "118"
```

Syntax check on update

- [root@localhost syslog-ng]# syslog-ng -s -f sudo.conf
- [2023-03-17T15:50:53.583894] WARNING: Configuration file format is too old, syslog-ng is running in compatibility mode. Please update it to use the syslog-ng 4.1 format at your time of convenience. To upgrade the configuration, please review the warnings about incompatible changes printed by syslog-ng, and once completed change the @version header at the top of the configuration file; config-version='3.37'
- [2023-03-17T15:50:53.591649] WARNING: \$(format-json) starts using type information associated with name-value pairs in syslog-ng 4.0. This can possibly cause fields in the formatted JSON document to change types if no explicit type hint is specified. This change will cause the type in the output document match the original type that was parsed using json-parser(), add --no-cast argument to \$(format-json) to keep the old behavior;

Log parsed and recreated by syslog-ng 4.X

```
{"cee":{"sudo":{"accept":{"uuid":"3d3e1f8c9b-d62e-4081-8fff-bc71e80e29","ttyname":"/dev/pts/0","submituser":"czanik","submithost":"localhost.localdomain","submitcwd":"/home/czanik","submit_time":{"seconds":1679063411,"nanoseconds":336097049,"localtime":"Mar 17 15:30:11","iso8601":"20230317143011Z"},"server_time":{"seconds":1679063413,"nanoseconds":696181185,"localtime":"Mar 17 15:30:13","iso8601":"20230317143013Z"},"runuser":"root","runuid":0,"runenv":["LANG=en_US.UTF-8","HOSTNAME=localhost.localdomain","TERM=xterm-256color","PATH=/home/czanik/.local/bin:/home/czanik/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin","MAIL=/var/mail/root","LOGNAME=root","USER=root","HOME=/root","SHELL=/bin/bash","SUDO_COMMAND=/usr/bin/ls /root/","SUDO_USER=czanik","SUDO_UID=1000","SUDO_GID=1000"],"runcwd":"/home/czanik","runargv":["ls","/root/"],"lines":60,"command":"/usr/bin/ls","columns":118}}},"app":{"name":"cee"},"SOURCE":"s_sys","PROGRAM":"sudo","PRIORITY":"notice","PID":"35167","HOST_FROM":"localhost","HOST_T":"localhost","FACILITY":"authpriv","DATE":"Mar 17 15:30:13"}}
```

Log parsed and recreated by syslog-ng 4.X

```
"runuser": "root",
"runuid": 0,
"runenv": [
  "LANG=en_US.UTF-8",
  "HOSTNAME=localhost.localdomain",
  [...],
  "SUDO_USER=czanik",
  "SUDO_UID=1000",
  "SUDO_GID=1000"
],
"runcwd": "/home/czanik",
"runargv": [
  "ls",
  "/root/"
],
"lines": 60,
"command": "/usr/bin/ls",
"columns": 118
```

How type support affects you?

- Only if you use PatternDB / JSON parsers (note: sudo JSON logs)
- Some destinations might reject / unable to process typed data
- Your alerts / reports / dashboards might change

Comparisons in *syslog-ng* 3.x

- Two different sets of comparisons operators
 - one to compare strings (eq, ne, gt, lt)
 - one to compare numbers (==, !=, >, <)
- Only integers are handled, no floats
- All values enclosed in quotes
- Easy to use the wrong operator and get unexpected results

Unexpected results

```
log {  
  source { file("/var/log/apache2/access.log"); };  
  parser { apache-accesslog-parser(); };  
  if ("${.apache.httpversion}" == "1.0") {  
    file("/logs/http-10-logs");  
  };  
};
```

- Integer comparison, so $1.X = 1.0$ for any X 😊

Comparisons in syslog-ng 4

- One set of comparison operators: ==, !=, >, <
- Type determined by quotes (strings use quotes, others do not)

```
@version: 4.0
```

```
log {
```

```
    source { file("/var/log/apache2/access.log"); };
```

```
    parser { apache-accesslog-parser(); };
```

```
    if ("${.apache.httpversion}" == 1.0) {
```

```
        file("/logs/http-10-logs");
```

```
    };
```

```
};
```

Python changes

- All your scripts work in v3 compatibility mode
- Includes and object declarations have to be changed
- Possibility to use Python virtualenv
 - Official distro packages use system Python packages
 - Unofficial packages use virtualenv, where Python dependencies are missing from distro

Python changes

v3

```
import socket
```

```
class SngResolver(object):  
    def parse(self,  
log_message):  
[...]
```

v4

```
import socket  
from syslogng import  
LogParser
```

```
class  
SngResolver(LogParser):  
    def parse(self,  
log_message):  
[...]
```

Kubernetes log enrichment

- 3.X could collect Kubernetes log messages
- 4.0 added support for Kubernetes log enrichment
 - Implemented in Python
 - Uses the Kubernetes API to fetch pod information to enrich log messages
 - It can cache information for better performance

For more syslog-ng 4 info

- Syslog-ng 4.0 release notes:
<https://github.com/syslog-ng/syslog-ng/releases/tag/syslog-ng-4.0.0>
- For Python support related changes:
<https://github.com/syslog-ng/syslog-ng/tree/master/modules/python-modules>

What is next?

- Improved monitoring support
 - Prometheus compatible data structures
 - More parameters can be monitored

Summary

- Most syslog-ng features arrived during the 3.X series
- Version 4 added:
 - Type support
 - More flexible Python support
 - Kubernetes log enrichment
- What is next?
 - Improved, Prometheus compatible monitoring support

 **ONE IDENTITY™**